A Next Generation IDE through Multi Tenant Approach

Sidhanth N., Sanjeev S., Swettha S., and Srividya R.

***Abstract*—The Integrated Development Environment (IDE) as a desktop application is often used for software development process. IDE's goal is to maximize the software productivity by providing developers with what they need all in one place. The program coding and its sharing has been indispensible in all kinds of software development; quite often the development involves team development where the interaction between the members and information regarding the status of the software development is crucial. But these IDE's are standalone applications hence they suffer several disadvantages for they need several software and hardware prerequisites.**

The idea of this paper is to present an IDE for next generation using cloud computing and providing the multi tenant service to software developers using SaaS. This proposed Online IDE is different from traditional desktop IDEs and has several impressive features making it outwit the traditional desktop IDEs of its kind.

***Index Terms*—Collaborative-working, e-learning, IDE, multi-tenancy, online-IDE, SaaS.**

I. Introduction

With the increasing demand for programming and the software developers provision for a perpetual Integrated Development Environment (IDE) application has become indispensible. There are several desktop IDEs like NETBEANS and ECLIPSE, which facilitates the programming environment with several advanced features. But these IDEs fail to meet the vital need of the software development which is "Resource (program) sharing". The file sharing becomes quite tedious when such desktop applications are used. Moreover these desktop applications demand installing and setting up of required environment, which are subjected to several hardware and software limitations.

Providing an IDE online via a browser as SaaS (Software as a Service) has several advantages and provides the programmers with immense facilities scoring more than the prevailing desktop IDEs. When an IDE environment is provided for the software developers as a service via a web browser the developer is not subjected to any hardware limitation for installing or deploying the IDE as in the case of desktop application, all he/she needs is a web browser. Moreover the files (or programs) of the developer can be shared across several developers, say a project team. The sharing of a single program or entire project is simple and comes implicit with the SaaS model. The developers can share the file in various levels of permissions such as edit, modify, append etc.

The tenants are distinguished by their tenant id, a user name and a password. The developers after logging into the proposed IDE via a web browser can just create and save files as they do in any desktop application. The developers are provided with a virtual desktop application like environment with which the developers can create folders and manage their files. These features match the perpetuate need of the software companies, as the project manager or the members can manage and keep track of the status of the project rather than looking for several individual files from the team members and integrating them, which many a times becomes a hectic process.

The developers who have an account can also use the project Uniform Resource Locator (URL) allocated to the project and share it with anyone across the globe by making the view permission of the file or folder as public. Every file or projects are allocated with a distinct URL link with which sharing is further more facilitated than ever.

The proposed online IDE includes some advanced features such as auto completion, auto indentation, braces pair matching, block comment, block uncomment, undo, incremental search, redo, find and replace. The proposed online IDE also provides the tenants with feature for theme customization and font size and font face changing. Thus the proposed online IDE outwits the desktop IDE for software development in much functionality concerned.

II. Related Works

Google produces Google Docs & Spreadsheets as in [1] and so on. These web-based softwares have gained huge attraction as the web-base software interaction is very similar to traditional desktop software, in fact they provide more flexibility. Some other industry companies such as EyeOS [2], CA Berkeley [3] have developed their Web OS for software usage. The proposed online IDE can be deployed in these WebOS often which has its own programming framework, to adapt to massive existing software. Some other software vendors try to provide web-based software focusing on certain function, such as Salesforce.com [4], which has produced a CRM online version; Comparing with these solutions, the SaaS system could easily adopt to the online IDE software. Virtual machines (VM) provide a good encapsulated and isolated environment for the software execution, but it needs strong capability to support VM running.

III. IDE Application Architecture, Data Extraction and Separation Component

The proposed online IDE solution is a Service Oriented

Manuscript received September 5, 2013; revised November 12, 2013. This work was supported by Computer Science Department, Sri Venkateswara College of Engineering, TamilNadu, India .

The authors are with Sri Venkateswara College of Engineering, TamilNadu, India (email: sidhanth.surana@gmail.com, ronniesmack@gmail.com, swetthacse@gmail.com, srividya94@gmail.com).

Architecture (SOA) as in [5], [6] based Software as a Service (SaaS) as in [7], [8], an application which supports multiple tenants as in [9], [10].

In the SaaS system, the proposed online IDE is executed and distributed on different physical or virtual machines. How developers interact with the presentation windows of those distributed execution software will be a big challenge. A virtual display layer is introduced in the proposed SaaS system to resolve the problem as in [11]. The main functions of this layer are: Virtual display instances management and Desktop windows merging.

The architecture of the online IDE on distributed file system is shown in Fig. 1. The developers interact with the web application server via web browser and are linked to the managed and unmanaged file or data storage server via the data access engines which are connected to all the remote servers and storage devices as described in [12].

The IDE supports serving multiple tenants at any instant of time. The tenant identification and directing him/her to their respective workspace or file location becomes essential. Each developer is allocated a separate log-in id which becomes the user directory in the cloud file system. The files and projects of those particular developers reside only in his/her allocated space facilitating file retrieval or file extraction from the storage components of the cloud. The developers can use the proposed online IDE by signing up, which creates an XML log data for that developer. The developer's programs and projects are stored in the cloud storage whose file information is added to the XML log data.

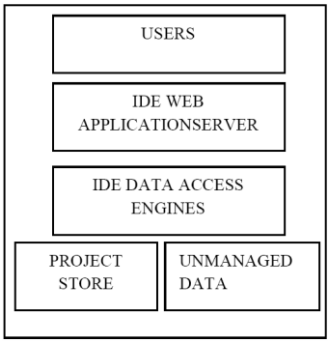

Fig. 1. Online IDE layered architecture.

Later when a file is opened, we can access the respective developer's files by getting information about the file of that particular developer.

IV. Multitenancy in The Proposed Online IDE

Multitenancy is the ability to allow each tenant to create custom extensions of the standard data objects. Tenants using a multitenant service operate in a virtual isolation from one another as in [13]. The proposed single online IDE application effectively morphs at run time for any particular tenant at any given time. The developers are treated as though each have a separate instance of the IDE application despite the fact that all the tenants work in a single set of hardware resources.

Multitenancy is practical only when it can support applications that are reliable, customizable, upgradeable, secure, and fast. For a multitenant application allows each tenant to create custom extensions to standard data objects and entirely new custom data objects and to keep tenant-specific data secured as in [14] in a shared database so that one tenant cannot see another tenant's data and the application should be more dynamic. Also when one tenant customizes the application's interface and business logic in real time without affecting the functionality or availability of the application for all other tenants we can bear the application's code base to be patched or upgraded without breaking tenant-specific customizations.

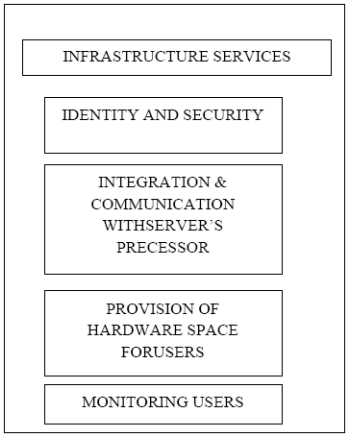

Fig. 2. SaaS key services in online IDE.

It's difficult to create a statically compiled application executable that can meet these and other unique challenges of multitenancy. Inherently, a multitenant application must be dynamic in nature, or polymorphic, to fulfill the individual expectations of various tenants and their users. For these reasons, multitenant application designs have evolved to use a runtime engine that generates application components from metadata about the application itself. In a well-defined metadata-driven architecture, there is a clear separation of the compiled runtime engine (kernel), application data, the metadata that describes the base functionality of an application, and the metadata that corresponds to each tenant's data and customizations. These distinct boundaries make it possible to independently update the system kernel, modify the core application, or customize tenant-specific components [15], [16], with virtually no risk of one affecting the others. The proposed online IDE application's response time scale even when tens of thousands of tenants subscribe to the service as it is developed on the multi tenant infrastructure.

V. Algorithm for Editor Syntax Highlight, Auto Indentation and Auto-File Extension

The IDE consists of an editor with rich features such as language recognition, keyword recognition, automatically indenting the text for easy readability.

As mentioned above, the proposed online IDE recognizes the language of the program in the text area and saves the file with the corresponding extension. The editor checks the word if it is a key word after every key press event. If the word is a keyword then it changes the color as referred from the XML file.

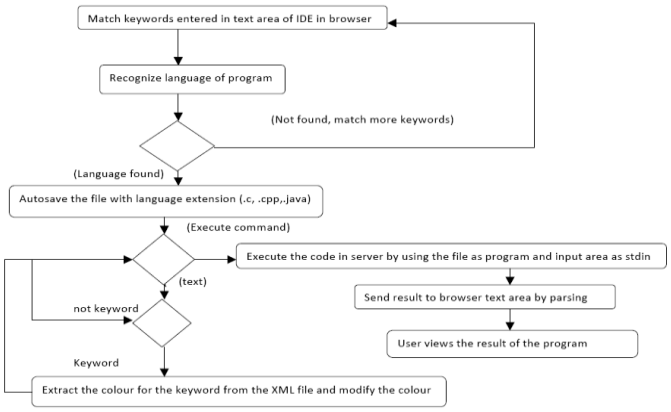

Fig. 3. The online IDE editor working algorithm.

If the run or compile button is pressed then the event for process execution is created and the corresponding program is executed. The output that is generated is directed from standard output to file which is later parsed and send to the output text area of the tenant's browser interface. The working mechanism is shown in Fig. 3, which handles the save, syntax highlight and the execution of the developer program.

Algorithm 1: For recognizing and coloring the keywords

```
1 Procedure keyword recognize (T, X)
2 T is the word entered by user
3 X is the XML file
4 begin
5   flag is set to 0
6   foreach Match keyword in X with T do
7     if T matches a keyword in XML file X then
8       Retrieve the corresponding colour for T
        from the XML file X
9     end
10  end
11 end
```

The above algorithm shows the procedure for reading the words from the input data and to check if they are a keyword or not. If they are a keyword then the word is correspondingly colored after referring from a XML file. These keywords are passed to language recognize function till flag equals zero.

Algorithm 2: Language recognize (T, X)

```
1 Procedure language recognize (T, X)
2 T is the keyword passed from keyword recognize
3 X is the XML file
4 begin
5   repeat
6     Match more keywords from the XML file X
7     if T == specific keyword in XML file X then
8       Auto save the file with recognized
        language extension(.c or .cpp)
9       return 1
10    end
11  until T != specific keyword from XML file X
12 end
```

The Algorithm 2: Language recognize is used to check the keyword passed from the keyword recognize routine with specific key-words from the XML file. If the language is found it auto saves the program with the recognized language extension (.c or .cpp). If the language is found it returns 1 which over writes the value of flag else it returns zero

VI. Implementation and User Interface

The IDE is designed according to the conventional design of an IDE so that the user can easily migrate to the proposed online IDE. The IDE consists of an editor as mentioned in Section A, a text area wherein the input can be given, the output area where the output of the program can be viewed and menu bar for performing operations such as save, compile, run, find and replace, cut, copy etc. The input and output text area is located below the editor. The Menu bar is located over the editor.

The user interface of the online IDE is developed using HTML5 and JavaScript. The server side programming is performed using JSP and the database used is XML-DB.

The tenants also have facility to customize their editor which will be saved and displayed accordingly when the same user logs in next time.

The input to the program is given in the text area below which will be used while running the program in the server and the result will be thrown back to the web browser in the text area below the editor.

The IDE is supported by all browsers like IE, Mozilla firefox, Safari, Chrome, opera, Netscape navigator etc. The IDE can be accessed from any device which has a web browser. Even mobile phones or PDAs can be used to reach the proposed online IDE once hosted as a service.

VII. Complexity

A linear search is done to recognize the keywords when the user gives the input data in the text area word by word. The keywords are matched with the keywords of the C/C++ to decide the language in which the user is writing. If the keywords matched cannot decide the language of the program(C/C++) then the linear search continues to match more keywords. Once the language is found the server auto saves the file with the corresponding extension(.c/.cpp).As the language is found by comparing just few but not all the keywords from the XML file the time complexity of the linear search is less than O(n) where n is the number of keywords.

VIII. Conclusion

The proposed online IDE supports jargon programming language built on SaaS architecture, with multi tenancy and it is likely to lead the world as the next generation IDE as it provides the developers with speed, comfort and convenience outwitting the traditional desktop IDEs. The IDE not only allows the users to access their programs anywhere but also enables them to collaborate with their team members. It also provides the developers with a number of

features that have been mentioned thus making the task of programming delightful with minimal hardware requirements. The facility to set themes and the desired fonts enables one to customize the IDE according to their whims and fancies.

The proposed online IDE supports jargon programming language built on SaaS architecture, with multi tenancy and it is likely to lead the world as the next generation IDE as it provides the developers with speed, comfort and convenience outwitting the traditional desktop IDEs. The IDE not only allows the users to access their programs anywhere but also enables them to collaborate with their team members. It also provides the developers with a number of features that have been mentioned thus making the task of programming delightful with minimal hardware requirements. The facility to set themes and the desired fonts enables one to customize the IDE according to their whims and fancies.

ACKNOWLEDGMENT

We thank our mentor R.Vanaja, Assistant Professor, C.S.E department, S.V.C.E for helping us throughout the project. We thank M.Sivanandham, principal, SV.C.E and B.Govindarajalu,H.O.D, C.S.E department, S.V.C.E for their support and encouragement. We thank K.S.Subhashini, , K.S.Gayathri, S.Senthamizh Selvi of the CSE department and P. Ganapathy of the Humanities and Social Science department for their support, help and encouragement. A special thanks to Sakshi jain for proof reading and our families for their support.

REFERENCES

[1] Google. [Online]. Available: http://documents.google.com
[2] Eyeos. [Online]. Available: http://eyeos.org/
[3] A. Vahdat, T. Anderson, M. Dahlin, D. Culler, E. Belani, P. Eastham, and C. Yoshikawa, "WebOS: Operating System Services For Wide Area Applications," *The Seventh IEEE Symposium on High Performance Distributed Computing*, July, 1998.
[4] Salesforce. [Online]. Available: http://www.salesforce.com
[5] L. Zhong, T. Wo, J. Li, and B. Li, "A Virtualization-based SaaS Enabling Architecture for Cloud Computing," presented at Sixth International Conference on Autonomic and Autonomous Systems, 2010.
[6] X. Yang, R. P. Bruin, M. T. Dove, A. Walkingshaw, T. V. Mortimer-Jones, R. Sinclair, D. J. Wilson, V. Milman, and T. Donovan, "A Service-Oriented Framework for Running Quantum Mechanical Simulations of Material Properties in a Grid Environment, " *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions* on Issue Date, July 2010.
[7] J. Zou, Y. Wang, and L. Kwei-Jay, "A Formal Service Contract Mode for Accountable SaaS and Cloud Services," *IEEE International Conference on Services Computing*, 2010
[8] L. Feng, G. Weiping, Z. Q. Zhao, and W. Chou, "SaaS Integration for software Cloud," presented at Cloud Computing (CLOUD*)*, 2010 IEEE 3rd International Conference, July 2010
[9] Z. Pervez, L. Sungyoung, and L. Young-Koo, "Multi-Tenant, Secure, Load Disseminated SaaS Architecture," presented at Advanced Communication Technology (ICACT), The 12th International Conference, Feb. 2010.
[10] E. J. Domingo, J. T. Nino, A. L. Lemos, M. L. Lemos, R. C. Palacios, and J. M. G. Berbis, "CLOUDIO: A Cloud Computing-Oriented Multi-tenant Architecture for Business Information Systems," presented at Cloud Computing (CLOUD), 2010 IEEE 3rd International Conference, July 2010.
[11] K. Lanju and L. Qingzhong, "A Metadata-driven Cloud Platform for Delivery of SaaS Applications," in *Proc. the IEEE, International Conference on Information and Automaton*, Shenzhen, Chinna June 2011
[12] T. Kwok, J. Laredo, and S.Maradugu, "A Web Services Integration toManage Invoice Identification, Metadata Extraction, Storage and Retrieval in a Multi-tenancy SaaS Application," *IEEE International Conference on e-Business Engineering*, 2008
[13] E. J. Domingo, J. T. Nio, and A. L. Lemos, "CLOUDIO: A Cloud Computing-oriented Multi-Tenant Architecture for Business Information Systems ," presented at IEEE 3rd International Conference on Cloud Computing, 2010
[14] Y. Badr, F. R. Biennier, and S. Tata, "The Integration of Corporate Security Strategies in Collaborative Business Processes," presented at IEEE Transactions on Service Computing, July – September, 2011
[15] X. Yang, P. R. Moore, and C. B. Wong, "A component-based software framework for product lifecycle information management for consumer products," *IEEE Trans. Consum. Electron*., vol. 53, pp. 1195, 2007 .
[16] X. Yang, P. R. Bruin, T. M. Dove, A. Walkingshaw, T. Mortimerjones, R.Sinclair, D. J. Wilson, V. Milman, and T. Donovan, "A Service-Oriented Framework for Running Quantum Mechanical Simulations of Material Properties in a Grid Environment," presented at IEEE Transaction on Systems, July 2010.

Sidhanth N. was born in Udhagamandalam on June 12, 1994. He is pursuing BE CSE in Sri Venkateswara college of Engineeering, Chennai, India. He is a VOLUNTEER in Bhumi,Chennai.Interested in cloud computing and web applications.

Sanjeev S. was born in Neyvelli on August 27, 1994. He is pursuing B.E CSE in Sri Venkateswara College of Engineering,Chennai,India. Interested in cloud computing and Web applications.

Swettha S. was born in Chennai on May 13, 1995. She is pursuing BE CSE in Sri Venkateswara College of Engineering, Chennai, India. Interested in Cyber crime and mobile softwares.

Srividya R. was born in chennai on March 31, 1994. She is pursuing BE CSE in Sri venkatewara College of Engineering, chennai, India. Her Interested in Web designing.